may also include a random access memory (RAM) device 608 and a read-only memory (ROM) device 610, and peripheral devices such as a floppy disk drive and a compact disk (CD) ROM drive (not shown) that also communicate with the CPU 602 over the bus 620 as is well known in the art.

[0096] While the invention has been described and illustrated with reference to exemplary embodiments, many variations can be made and equivalents substituted without departing from the spirit or scope of the invention. Accordingly, the invention is not to be understood as being limited by the foregoing description, but is only limited by the scope of the appended claims.

[0097] What is claimed as new and desired to be protected by Letters Patent of the United States is:

1219776 v2; Q56_021.DOC

1. A datapath for processing input data, said datapath comprising:

at least one arithmetic pipeline, each pipeline inputting at least a portion of the input data and being controllable to perform at least one mathematical operation on the portion as it passes through the pipeline, each pipeline being capable of performing a four component dot product as the input data passes through said pipeline a single time.

2. The datapath of claim 1, wherein at least one pipeline is subdivided into a plurality of subsections, each subsection corresponding to a subset of the input data.

3. The datapath of claim 2, wherein at least one of said subsections comprises:

a floating point multiplier module; and

a flat four-input floating point adder module.

4. The datapath of claim 3, wherein said floating point multiplier module inputs the portion of the input data and performs a floating point multiply operation, and said flat four-input floating point adder module performs a normalization operation on a result of the floating point multiply operation.

1219776 v2; Q56_021.DOC

5. The datapath of claim 3, wherein said subsection further comprises a floating point-to-integer converter module controllable to convert a floating point number into one of a plurality of integer types and sizes.

6. The datapath of claim 2, wherein said plurality of subsections comprise:

a floating point multiplier module; and

a two-input floating point adder module.

7. The datapath of claim 6, wherein said floating point multiplier module inputs the portion of the input data and performs a floating point multiply operation, and said two-input floating point adder module performs a normalization operation on a result of the floating point multiply operation.

8. The datapath of claim 1, wherein said at least one mathematical operation is a three component dot product that is performed as the input data passes through said pipeline a single time.

9. A processor executing arithmetic operations on vertex data, said processor comprising:

a data processing engine coupled to a first stage of said graphics pipeline, said processing engine comprising at least one arithmetic pipeline, each pipeline inputting at least a portion of the input data and being controllable to perform at least one mathematical operation on the portion as it passes through the pipeline,

1219776 v2; Q56_02!.DOC

each pipeline being capable of performing a four component dot product as the input data passes through said pipeline a single time.

10. The processor of claim 9, wherein at least one arithmetic pipeline is subdivided into a plurality of subsections, each subsection corresponding to a subset of the input data.

11. The processor of claim 10, wherein at least one of said subsections comprises:

a floating point multiplier module; and

a flat four-input floating point adder module.

12. The processor of claim 11, wherein said floating point multiplier module inputs the portion of the input data and performs a floating point multiply operation, and said flat four-input floating point adder module performs a normalization operation on a result of the floating point multiply operation.

13. The processor of claim 12, wherein said subsection further comprises a floating point-to-integer converter module controllable to convert a floating point number into one of a plurality of integer types and sizes.

14. The processor of claim 10, wherein said plurality of subsections comprise:

a floating point multiplier module; and

1219776 v2; Q56_02!.DOC

a two-input floating point adder module.

15. The processor of claim 14, wherein said floating point multiplier module inputs the portion of the input data and performs a floating point multiply operation, and said two-input floating point adder module performs a normalization operation on a result of the floating point multiply operation.

16. The processor of claim 9, wherein said at least one mathematical operation is a three component dot product that is performed as the input data passes through said arithmetic pipeline a single time.

17. A graphics pipeline comprising:

a vertex engine coupled to a first stage of said graphics pipeline, said vertex engine comprising a plurality of datapaths, each datapath inputting vertex data and being controllable to perform at least one mathematical operation on the vertex data as the data passes through the datapath, wherein each datapath is capable of performing a three component dot product as the vertex data passes through said datapath a single time.

18. The graphics pipeline of claim 17, wherein each datapath is capable of performing a four component dot product as the vertex data passes through said datapath a single time.

19. The graphics pipeline of claim 17, wherein each datapath is a multi-function floating point pipeline.

1219776 v2; Q56_021.DOC

20. An arithmetic pipeline comprising:

a floating point to integer converter module, said module being controllable to convert a single precision floating point number to an integer number of selectable bit-sizes and format.

21. An arithmetic pipeline comprising:

a two input adder module, said module being controllable to add a first single precision floating point number to a second single precision floating point number and to output a resulting single precision floating point number, said module comprising:

means for inputting mantissa portions of the first and second single precision floating point numbers, said mantissa inputting means determining a larger number and a smaller number, and outputting a mantissa portion of the larger number and a mantissa portion of the smaller number;

means for inputting exponent portions of the first and second floating point numbers, said exponent portions inputting means determining and outputting a larger exponent;

means for inputting sign-bits of the first and second floating point numbers, said sign-bits inputting means determining and outputting a sign-bit for said resulting floating point number;

carry-in generation means for outputting carry-in data based on sign-bits of the first and second floating point numbers and the mantissa portion of the larger number;

1219776 v2; Q56_02!.DOC

addition logic receiving the carry-in data, mantissa of the larger number, mantissa of the smaller number, and a difference between the larger and smaller exponents, said addition logic shifting the mantissa of the smaller number to align with the mantissa of the larger number, calculating and outputting a normalized mantissa output and exponent modifier; and

output logic receiving the sign-bit result, the normalized mantissa output and the exponent modifier, said output logic outputting the resulting single precision floating point number based on the normalized mantissa output and exponent modifier.

22. The arithmetic pipeline of claim 21, wherein all arithmetic negations are approximated to a logical negation and said carry-in generation means generates the carry-in data to correct the approximations.

23. The arithmetic pipeline of claim 21, wherein said carry-in generation means generates the carry-in data to correct any loss of precision that may have occurred in shifting of the mantissa of the smaller number.

24. The arithmetic pipeline of claim 21, wherein said carry-in generation means generates the carry-in data to correct incorrect determinations of which floating point number is larger.

1219776 v2; Q56_021.DOC

25. The arithmetic pipeline of claim 21, wherein said carry-in generation means generates the carry-in data to correctly round the resulting single precision floating point number to meet IEEE 754 rounding mode rules.

26. An arithmetic pipeline comprising:

a flat four-input single precision floating point adder module, said module being controllable to add first, second, third and fourth single precision floating point numbers and to output a resulting single precision floating point number, said module comprising:

means for predicting a largest number from exponent and mantissa portions of said floating point numbers, said predicting means outputting a plurality of shifting data calculated based on said largest number and said exponent portions;

means for partially sorting said floating point numbers based on sign-bit and the exponent portions of said floating point numbers, said sorting means outputting sorted mantissas, sorted exponents, and sorted sign-bits;

carry-in generation means for outputting carry-in data based on said sorted sign-bits and mantissas;

addition logic receiving the carry-in data and said sorted mantissas and said plurality of shifting data, said addition logic calculating and outputting a normalized mantissa output and exponent modifier; and

output logic receiving the normalized mantissa output, exponent modifier, and a largest exponent, said output logic outputting the resulting floating point

1219776 v2; Q56_02!.DOC

number based on the normalized mantissa output, the exponent modifier, and the largest exponent.

27. The arithmetic pipeline of claim 26, wherein all arithmetic negations are approximated to a logical negation and said carry-in generation means generates the carry-in data to correct said approximations.

28. The arithmetic pipeline of claim 26, wherein said carry-in generation means generates the carry-in data to correct any loss of precision that may have occurred in shifting of non-largest mantissas by said addition logic.

29. The arithmetic pipeline of claim 26, wherein said carry-in generation means generates the carry-in data to correct incorrect determinations of which floating point number is larger.

30. The arithmetic pipeline of claim 26, wherein said carry-in generation means generates the carry-in data to correctly round the resulting single precision floating point number to meet rounding mode requirements.

31. The arithmetic pipeline of claim 26 further comprising a floating point multiplier module, said multiplier module inputs the input data and performs a multiply operation and said four-input single precision floating point adder module performs a normalization operation on a result of the multiply operation.

1219776 v2; Q56_021.DOC

32. A processor system comprising:

a processor; and

a data processing pipeline coupled to said processor, said data processing pipeline comprising at least one datapath, each datapath inputting data and being controllable to perform at least one mathematical operation on the data as the data passes through the datapath, wherein each datapath is capable of performing a four component dot product as the data passes through said datapath a single time.

33. A processor system comprising:

a processor; and

a graphics point pipeline coupled to said processor, said graphics pipeline comprising a vertex engine coupled to a first stage of said pipeline, said vertex engine comprising a plurality of datapaths, each datapath inputting vertex data and being controllable to perform at least one mathematical operation on the vertex data as the data passes through the datapath, wherein each datapath is capable of performing a three component dot product as the vertex data passes through said datapath a single time.

34. The system of claim 33, wherein each datapath is capable of performing a four component dot product as the vertex data passes through said datapath a single time.

1219776 v2; Q56_02I.DOC